This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

A New Deep-Q-Learning-Based Transmission Scheduling Mechanism for the Cognitive Internet of Things

Jiang Zhu, Yonghui Song, Dingde Jiang, *Member, IEEE*, Houbing Song, *Senior Member, IEEE*

Abstract—**Cognitive networks (CNs) are one of the key enablers for the Internet of Things (IoT), where CNs will play an important role in the future internet in several application scenarios, such as healthcare, agriculture, environment monitoring, and smart metering. However, the current low packet transmission efficiency of IoT faces a problem of the crowded spectrum for the rapidly increasing popularities of various wireless applications. Hence, the IoT that uses the advantages of cognitive technology, namely the cognitive radio-based Internet of Things (CIoT), is a promising solution for IoT applications. A major challenge in CIoT is the packet transmission efficiency using CNs. Therefore, a new Q-learning-based transmission scheduling mechanism using deep learning for the CIoT is proposed to solve the problem of how to achieve the appropriate strategy to transmit packets of different buffers through multiple channels to maximize the system throughput. A Markov decision process based model is formulated to describe the state transformation of the system. A relay is used to transmit packets to the sink for the other nodes. To maximize the system utility in different system states, the reinforcement learning method, i.e., the Q learning algorithm, is introduced to help the relay to find the optimal strategy. In addition, the stacked auto-encoders deep learning model is used to establish the mapping between the state and the action to accelerate the solution of the problem. Finally, the experimental results demonstrate that the new action selection method can converge after a certain number of iterations. Compared with other algorithms, the proposed method can better transmit packets with less power consumption and packet loss.**

***Index Terms*—Internet of Tings; cognitive networks; Markov decision process; Q learning; deep learning**

I. Introduction

In the future, wireless sensor networks are expected to be integrated into the Internet of Things [1-2], where reconfigurable, flexible, and intelligent sensors dynamically join the Internet and use it to collaborate and accomplish their tasks for a wide range of applications in various domains [3-8], such as big data applications, Internet of Things, E-commerce, medical device [9-10], virtual reality & augmented reality, and environment monitoring. The network environment also tends to become increasingly complicated, and the communication resources become increasingly scarce. It is a great challenge to the wireless sensor networks and Internet of Things. Coincidentally, the cognitive network technology can compensate for these deficiencies [11-15]. Cognitive nodes are intelligent wireless devices that can sense the environment, observe the network changes, use the knowledge learnt from the previous interaction with the network, and make intelligent decisions to seize the opportunities to transmit. The process of continuously sensing the environment information, exchanging control information, learning information, deciding and executing a strategy in the network can provide the ability of intelligence and adaptability to the wireless sensor networks and the future Internet of Things. Therefore, the cognitive radio technology is a key communication approach for resource-constrained wireless sensor networks and future wireless network [16-20]. When cognitive users, i.e., sensors in wireless sensor networks, access the spectrum, to effectively use the network resources and satisfy the throughput demand for multimedia applications, effective mechanisms are required to coordinate the actions of the cognitive users (transmission power control, spectrum access, transmission scheduling, et al.) [21-24]. With the rapid increase in number of wireless devices in the Internet of Things, more data will be stored in the network nodes. Thus, the method to rapidly forward data with the limited storage space and bandwidth is a great challenge for the current wireless network of Internet of Things.

Currently, many existing literatures (see in [25]-[33]) have studied the problem of network data transmission with unknown environment information in a cross-layer design manner. Among these literatures, [25] and [26] have examined the adaptive modulation algorithm in the data transmission stage, whereas [27] focused on the reliable route discovery to reduce the data online time. [28] proposed the quality of service (QoS) awareness scheduler and power adaptation scheme at both uplink and downlink medium access control (MAC) layers

This work is supported by the National Nature Science Foundation of China (No. 61102062, 61271260, 61571104, 61572231) and Key Project of Chinese Ministry of Education (No.212145), the Nature Science Foundation of Chongqing Science and Technology Commission (No.cstc2015jcyjA40050), the Science and Technology Research Project of Chongqing Education Commission (No. KJ120530), the Fundamental Research Funds for the Central Universities (No. N150402003). Both D. Jiang and H. Song are corresponding authors.

J. Zhu, Y. Song and Y Liu are with Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: zhujiang@cqupt.edu.cn).

D. Jiang is with the School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China (e-mail: jiangdd@uestc.edu.cn).

H. Song is with the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: h.song@ieee.org).

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

to coordinate the action of the lower layers for resource efficiency. [29] focused on the throughput and fairness. Depending on the preference of the two features, algorithms and methods are proposed to assign or schedule users to prioritize to maximize throughput, maximize fairness or finding the appropriate balance between the two. In [30], a cross layer design of MAC and routing protocols and the topology design are studied to gain more profits for networking. In addition, [31-33] modeled the optimization problem of wireless networks as Markov decision process (MDP) to describe the state transformation of the system. However, it is difficult to solve the MDP problem because the MDP has many variables. Therefore, the reinforcement learning method can be introduced to solve it. [33-37] research the scheduling for different applications or services in distributed wireless networks. The scheduling mechanisms are in cross-layer way, and quality of service (QoS) or quality of experience (QoE) are involved in the designs of scheduling mechanisms.

For the problem of the curse of dimensionality, a new Q-learning-based transmission scheduling mechanism using deep learning is proposed for the MDP problem to intelligently make the appropriate strategy. The model formulated in this paper considers the power required for packet transmission in the wireless networks and the packet loss to maximize the system throughput. We modify the action selection by the comprehensive action evaluation method based on the *Q* value and index value to consider the balance of exploration and exploitation. In addition, when the system scale is large, the state space and action space of the system are also notably large, and it is notably difficult to calculate the optimal action corresponding to the state one by one. Therefore, we use the method of deep learning to construct the mapping between state and action to quickly obtain the strategy.

The contributions of the paper are as follows:

1) The MDP model is formulated to describe the problem of transmission scheduling for the cognitive Internet of Things;

2) The Q learning algorithm is modified to learn the system state transition without the system prior information;

3) The stacked auto-encoder deep learning algorithm is adopted to map the relation between the states and the actions to avoid the massive calculation and storage in the Q learning phase.

This paper is organized as follows. Sections II and III introduce the system model and MDP problem. Then, section IV introduces the proposed scheme of this paper, i.e., the deep Q learning algorithm, and the algorithms are compared in section V. Section VI provides the scheme simulation and analysis. Finally, we summarize the paper and point out the future direction. Some notations are listed in Table 3 for the reader's convenience.

II. SYSTEM MODEL

As shown in Fig. 1, a CIoT that coexists with a licensed system is considered. In the system, a point-to-point transmission from the relay to the sink and *M* frequency domain channels are considered. One relay gathers packets from its *K* neighbor nodes, and these packets are stored in *K* buffers with identical length *L*. The packets that come from *K* neighbor nodes are assumed to have Poisson distribution with identical arrival rate λ . *M* channels are independent and identically distributed (i.i.d.). The transmission schedule is decided by the relay. In a frame, the relay selects a channel and transmits packets for a node to the sink. When the channel state is poor, the relay does not transmit packet. When a certain buffer is full, if the relay does not transmit packet for it, then, the packet are lost if packets continue to arrive in the next frame. Therefore, in the packet transmission process, the relay must comprehensively consider the channel state and the buffer states that correspond to the communication pairs and transmission mode.

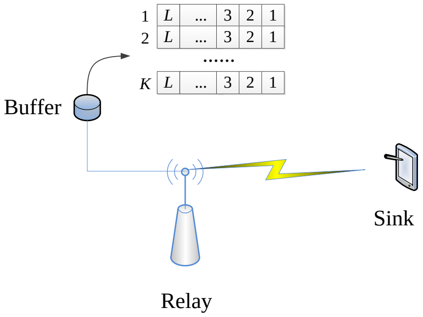

Fig. 1. System model

A. *Channel state*

In the system model, the time unit is defined as frame T_f , the state of the channel does not change in each frame, and the state transition of the channel occurs between two adjacent states. The channel state can be modeled as a finite-state Markov chain [38]. The signal-to-noise ratio (SNR) is assumed to obey Rayleigh distribution, and the probability density function is expressed as $p(\rho)=1/\overline{\rho}\exp(-\rho/\overline{\rho})$, where $\rho>0$; $\overline{\rho}=E(\rho)$ is the average SNR. The threshold of the SNR is $\rho_{snr}=\{\rho_1,\rho_2,\cdots,\rho_{C-1}\}$, and *C* is the number of channel states. Then, we obtain the channel state space $C\triangleq\{c_0,c_1,\cdots,c_{C-1}\}$. Therefore, the probability distribution of channel state is

$$p_C(c_n)=\int_{\rho_n}^{\rho_{n+1}}p(\rho)d\rho \qquad (1)$$

The state transition probability of the channel is

$$p_C(c_n,c_{n+1})=N(\rho_{n+1})T_f/p_C(c_n),n\in\{1,2,\cdots,N-2\} \qquad (2)$$

$$p_C(c_n,c_{n-1})=N(\rho_n)T_f/p_C(\mathrm{c}_n),\quad n\in\{1,2,\cdots,N-1\} \qquad (3)$$

where $N(\rho_n)=\sqrt{2\pi\rho_n/\overline{\rho}}\,f_d\exp(-\rho/\overline{\rho})$; f_d is the maximum Doppler shift. Since *M* channels are i.i.d., the state transition probability of *M* channels is $p_C(c,c^{'})=\prod_{m=1}^{M}p_{c,m}(c_i,c_{i+1})$.

B. *Buffer state*

In each system frame T_f , the arriving packet obeys Poisson distribution with arrival rate λ , which is expressed as $p_{d_i}(d_i)=\exp(-\lambda_{d_i}T_f)(\lambda_{d_i}T_f)^{d_i}/d_i\,!$, where d_i is the number of arriving packets in each frame. At the beginning of frame i , to buffer k , the existing buffer length is $l_{i,k}$. If the number of

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

arriving packets is $d_{i,k}$ and the number of transmitted packets is $t_{i,k}$, then the new buffer length is

$$l_{k,i+1} = \min(l_{i,k} + d_{i,k} - t_{i,k}, L) \tag{4}$$

If the state transition probability of buffer k is $p_{l_k}(l_k, l_k^{'})$, the state transition probability of the K buffers is $p_l(l, l^{'}) = \prod_{k=1}^{K} p_{l,k}(l_i, l_{i+1} | a_{i,k})$, where $a_{i,k}$ is the number of packets transmitted for buffer k during frame i.

C. Transmission Power

To truly improve the transmission efficiency, the adaptive modulation (AM)[39] method is used to adjust the transmission power and rate. We use $j \in \{0,1,2,\mathrm{L}\ ,J\}$ to indicate the selected mode. 0 and 1 are corresponding to no transmission and BPSK transmission respectively, and $j \geq 2$ is corresponding to 2^j-QAM transmission. Given transmission rate, power, and channel state, the bit error rate (BER) can be estimated. Assuming ideal coherent phase detection, BER bounds are given by [40],

If j =1, then

$$p_{BER}(c_i, j) \leq 0.5erfc(\sqrt{\rho_i P(c_i, j)/WN_0}) \tag{5}$$

If j >1, i.e., $j = 2,3,\mathrm{L}$, then

$$p_{BER}(c_i, j) \leq 0.2\exp(-1.6\rho_i P(c_i, j)/WN_0(2^j - 1)) \tag{6}$$

where WN_0 is the noise power. The BER inequalities above give a pessimistic minimum power $P(c_i, j)$ to achieve a specified BER for channel state c_i and selected mode j.

III. MDP ANALYSIS OF TRANSIMISSION

The system contains two state objects: buffer state and channel state. The system operation is a process of state transition. The next system state is obtained by selecting and implementing a certain action at the current system state. Therefore, the state of the system in the next frame is only related to the current state and action. Thus, we model the transmission scheduling problem as Markov decision process.

A. Action Set

When state transition occurs, the relay must choose an action according to current state. The possible action of relay can be defined as $a_i \in A = \{a_{m,k,j}\}$, where $m \in \{1,2,\mathrm{L}\ ,M\}$, $k \in \{1,2,\mathrm{L}\ ,K\}$, $j \in \{0,1,2,\mathrm{L}\ ,J\}$. $a_i = a_{m,k,j}$ indicates that at the beginning of frame i , the relay selects channel m to transmit $a_{m,k,j}$ packets by selected mode j for buffer k.

B. State Transition Probability

The system state is the combination of the buffer state and channel state, i.e., $S @ B \otimes C$. If the buffer length is L, the number of buffer states of a single buffer is B=L+1. The number of channels is M.

To one buffer and one channel, the state transition probability is $p_l(l_i, l_{i+1} | a_i) \times p_c(c_i, c_{i+1})$. Therefore, the entire system state transition probability expression is

$$p_s(s_i, s_{i+1} | a_i) = \prod_{k=1}^{K} p_{l_k}(l_i, l_{i+1} | a_i) \times \prod_{m=1}^{M} p_{c_m}(c_i, c_{i+1}) \tag{7}$$

C. Utility

The goal of the paper is to maximizing the system utility. If the coding rate is V, the throughput (bits per symbol) under different transmission mode is $V \times j$. Thus, at system $s_i = \{l_i, c_i\} \in S$, the benefit of system after taking action a_i is

$$B(s_i, a_i) = \sum_{k=1}^{K} V \times j \tag{8}$$

where $R(s_i, a_i)$ is the throughputs when the system state is s_i , and action a_i is selected in frame i . It is mentioned in above section that adaptive transmission scheme based on M-QAM is employed for each channel. When $a_{m,k,j}$ packets are transmitted through channel m , corresponding transmission mode j is employed. Therefore, there is a mapping $\varphi(\cdot)$, that $\varphi(j) = a_{m,k,j}$. For simplification, we assume that $V \times j = a_{m,k,j}$.

The pressure value of buffer k is defined as $f_{k,i} = \exp(\theta \times l_{i,k})$, where θ is the pressure coefficient, which represents the number of packets in the buffer. When the number of packets in buffer increases, the arriving packets in the next frame may be lost because of the small space. Therefore, a smaller pressure value corresponds to less packets loss. Thus, the pressure value is inversely proportional to the performance of the system. In addition, if we have the constraint of the BER, we can obtain the minimum power transmission power $p_{s_i}(s_i, a_i)$ [41], which is also inversely proportional to the performance of the system.

Consequently, the system cost is the combination of buffer pressure and power consumption.

$$C(s_i, a_i) = (\sum_{i=1}^{K} f_{k,i}) \times p_{s_i}(s_i, a_i) \tag{9}$$

The system utility O_i is proportional to the number of transmitted packets in each frame and inversely proportional to the buffer pressure and power consumption. Hence, the system utility can be described as

$$O_i = O(s_i, a_i) = B(s_i, a_i)/C(s_i, a_i) \tag{10}$$

IV. DEEP-Q-LEARNING-BASED OPTIMAL ACTION ACQUISITION SCHEME

In the transmission process, the relay obtains the environment state information by learning to guide its action. The state and action of the system are discrete, and the state of the system discontinuously changes when an action is executed in a frame. Accordingly, to solve the MDP problem, the method of reinforcement learning is used to guide the action of node [42-45]. The Q learning (QL) algorithm learns the environment state information and obtains the optimal action. The QL algorithm is a type of gradual optimization process, so it is difficult to achieve fast convergence in action selection. The artificial neural network can compensate the limitation for its generalization and function approximation ability. In addition,

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

the deep learning method has proven good performance in many fields [46-48]. Thus, we use the deep learning method to establish a mapping between states and actions [49-53].

A. Q Learning Algorithm

In the process of the Q learning algorithm, the agent finds the optimal action by continuous interaction with the environment in a constant trial-and-error manner. The optimal action is related to immediate reward and considers the reward of future n steps. We use $V^{\pi}(s_i)$ to represent the entire reward it in Q learning algorithm under a policy π .

$$V^{\pi}(s_i) = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots \qquad (11)$$

In the Q learning algorithm, the Q value, which is the evaluation of the state and action, is the combination of immediate and discounted reward and can be formulated as

$$Q(s_i, a_i) \leftarrow r_i + \gamma V^{\pi}(s_{i+1}) \qquad (12)$$

where $\gamma(0 < \gamma < 1)$ is the discount coefficient, which indicates the effect of future reward on the current action. The learning goal of Q learning is to maximize the total utility. Thus, in (12), we replace r_i and $V^{\pi}(s_{i+1})$ with O_i and $\max_{a_{i+1}\in A} Q(s_{i+1}, a_{i+1})$, respectively.

$$Q(s_i, a_i) \leftarrow O_i + \gamma \max_{a_{i+1}\in A} Q(s_{i+1}, a_{i+1}) \qquad (13)$$

where A is the action sets.

In the learning stage, how to balance the exploration and exploitation of the action set is the key problem in Q learning. In particular, when the state of the system is large, how to effectively choose the action will directly affect the algorithm convergence and system performance. Therefore, to achieve the optimal action, we add a modified index value to quickly find the best action. The index value can reflect the fluctuations of the rewards and timely adjust the explore range to reduce the unnecessary selection cost.

$$a_i \leftarrow \arg\max_a (Q(s_i, a) + Index(s_i, a)) \qquad (14)$$

where Q is the evaluation value to the current state and action. Based on the Q value, $Index(s_i, a)$ is introduced to obtain the optimal potential action. Its expression is (15).

$$Index(s_i, a) = C_p \sqrt{2\ln i \times \min\{1/4, V_a(i)\}/T_a(i)} \qquad (15)$$

where C_p is a constant greater than 0 [54]. $T_a(i)$ is the number of times action a has been selected after i frames. $V_a(i)$ is the bias factor, which contains the utility value variance $\sigma_a^2(i)$ of the action to reflect its volatility.

$$\sigma_a^2(i) = \sum_{k=1}^{T_a(i)} O^2\left(s_k, a\right)/T_a(i) - O^2\left(s_{T_a(i)}, a\right) \qquad (16)$$

$$V_a(i) = \sigma_a^2(i) + \sqrt{2\ln i / T_a(i)} \qquad (17)$$

On one hand, the action selection method based on the action index considers the system utility of the current action and gradually considers the action with a larger effect, which reflects the characteristic of exploitation of the system. On the other hand, with the ongoing iterative process, if a certain action is not selected or the selected number is notably small, then it is biased to select the action in the next iteration, which reflects the characteristic of exploration.

After determining the execution action, the relay performs action a_i , calculates utility value O, and updates the Q value according to formula (18).

$$Q_{i+1}(s_i, a_i) = \begin{cases} (1-\alpha)Q_i(s_i, a_i) + \alpha(O_i + \gamma \max Q_i(s_{i+1}, a_{i+1})) \\ \qquad\qquad if \;\; s = s_i \;\; and \;\; a = a_i, \\ Q_i(s_i, a_i) \qquad\qquad otherwise. \end{cases} \qquad (18)$$

where $\alpha(0 < \alpha \le 1)$ is the learning rate of the state action and is calculated by $\alpha = 1/(1 + T_a(i))$.

The specific implementation process of the modified Q learning algorithm is shown below.

```
The Modified Q Learning Algorithm
Initialize action visiting number $T_i(n) = 0$ .
Initialize state-action value $Q(s_i, a_i) = 0$ and state-action
    look-up table.
for episode1=1, $I_1$ do
    Initialize action vector $a = \{a_1, a_2, \cdots\}$ .
    To the current state $s_i$ .
    for episode2=1, $Ite$ do
        if episode2=1
            Select a random action $a_i$ .
        end if
        if episode2>1
            $Index(s_i, a) \leftarrow C_p \sqrt{\frac{2\ln i}{T_a(i)} \min\{\frac{1}{4}, V_a(i)\}}$
            Select action according to the following formula
                $a_i \leftarrow \max_a (Q(s_i, a) + Index(s_i, a))$ .
        end if
        Execute $a_i$ and obtain $O_i$ and turn into state $s_{i+1}$ .
        Calculate $\alpha \leftarrow 1/(1 + T_a(i))$ .
        Update $Q(s_i, a_i)$ .
            $Q_{i+1}(s_i, a_i) \leftarrow \begin{cases} (1-\alpha)Q_i(s_i, a_i) + \alpha(O_i + \gamma \max Q_i(s_{i+1}, a_{i+1})) \\ if \; s = s_i \; and \; a = a_i, \\ Q_i(s_i, a_i) \quad otherwise. \end{cases}$
        Update state-action look-up table.
    end for
end for
```

B. Convergence Analysis of Algorithm

In the convergence analysis process of algorithm, the optimal Q value is $Q^*(s_i, a_i)$.

Theorem 1:

The value of the system utility function O_i , which is defined by formula (10) is bounded.

Proof:

Formula (10) consists of three parts: benefit and cost. The expression of benefit is $\sum_{k=1}^{K} V \times j$, which indicates the number of transmitted packets in a frame, is a finite value. The denominator is the cost function (9) which is the combination of buffer pressure and power consumption. The number of packets in the buffer and power consumption are finite. Consequently, the system utility is bounded. ■

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

Theorem 2:

Learning rate satisfiers that $0<\alpha\le 1$, and

$$\sum_{i=1}^{\infty}\alpha=\infty,\sum_{i=1}^{\infty}\alpha^2<\infty,\forall s,a \quad (19)$$

Proof:

Since $\alpha=1/(1+T_a(i))$ and $T_a(i)$ is the number of times action a has been selected after *i* frames, $0<\alpha\le 1$.

According to the definition of $T_a(i)$, $1/(1+T_a(i))<1/i$. It is clear that $\sum_{i=1}^{\infty}(1/i)^2=0<\infty$, therefore, $\sum_{i=1}^{\infty}\alpha^2<\infty$.

If each state-action pair are visited infinitely, It is obviously that

$$\sum_{i=1}^{\infty}\alpha=\sum_{i=1}^{\infty}1/(1+T_a(i))>\sum_{i=1}^{\infty}1/i \quad (20)$$

Since $\sum_{i=1}^{\infty}1/i=\infty$, $\sum_{i=1}^{\infty}\alpha=\infty$. ■

Theorem 3: [55]

Given bounded utility function O_i, learning rate $0<\alpha\le 1$, and $\sum_{i=1}^{\infty}\alpha=\infty$, $\sum_{i=1}^{\infty}\alpha^2<\infty$, $\forall s,a$, then

$$\lim_{i\to\infty}Q_i(s,a)=Q^*(s,a),\ \forall s,a \quad (21)$$

with probability 1. $Q^*(s,a)$ is Q value under the optimal policy π^*

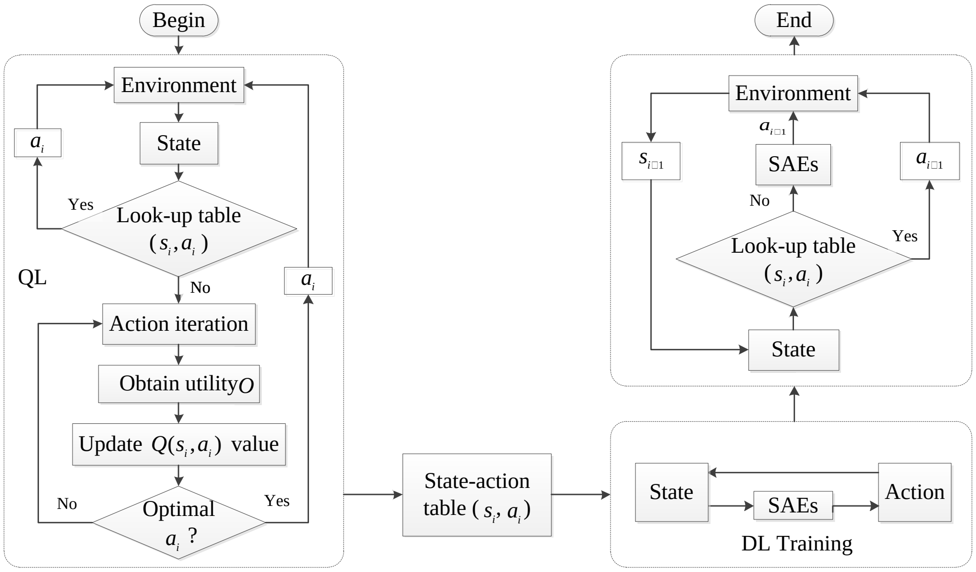

Fig. 3. Flow chart of the proposed scheme

C. Deep Action Mapping Network

The stacked auto-encoder (SAE) model of the deep learning is used to build the relations of states and actions. The structure of the model is shown in Fig. 2.

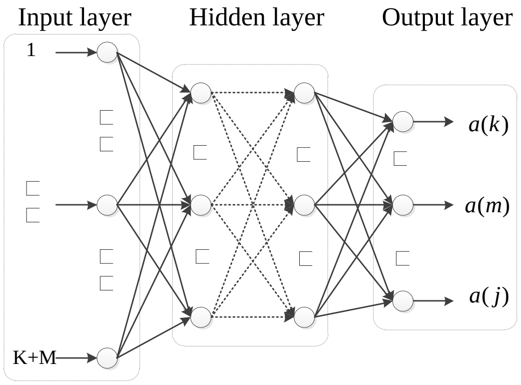

Fig. 2. Structure of the SAE model

The input layer of the model represents the state information of the system, and the number of neurons in the layer is *K+M*. The input vector can be expressed as $Input=[l_1,\cdots,l_k,\cdots,l_K,c_1,\cdots,c_m,\cdots,c_M,]$. The output layer, which represents the selection action information, can be expressed as $Output=[a(k),a(m),a(j)]$. It consists of the selected channel *m*, transmission mode *j* and buffer *k*. Its neuron number is *K+M+J*. The hidden layer consists of multiple layers. According to the previous experience, the number of neurons in hidden layer is

$$n_h=\sqrt{n_i+n_o}+Con \quad (22)$$

where n_i is the number of input layers, n_o is the number of output layers, and n_h is the number of hidden layers. *Con* is a constant limited in $[1,10]$.

The SAE model uses the logistic sigmoid function as the transfer function in the encoding and decoding process. The cost function $L(X)$ is defined as follows.

$$L(X)=\underset{x\in(0,1)}{\arg\min}\sum_{i=1}^{N}\left\|x^i-f(x^i)\right\|^2/2+\mu J_{W^l} \quad (23)$$

where $J_{W^l}=\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(\mathrm{W}_{ji}^l)^2$ is the weight decay term, whose function is to reduce the range of the weight to prevent over-fitting in the training section. The rules of weight and bias vector updating are as follows.

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

$$W(k+1) = W(k) - \beta \times \partial L / \partial W \tag{24}$$

$$b(k+1) = b(k) - \beta \times \partial L / \partial b \tag{25}$$

where β is the learning rate. The partial derivative of the cost function to the weight and bias parameters are calculated in formulas (26) and (27).

$$\begin{aligned} \partial L / \partial W &= -(y'_{m+1,i} - y_{m+1,i}) \times f'_m \times y_{m,i} + \mu W^m_{j,i} \\ &= -\delta^{m+1}_j \cdot y_{m,i} + \mu W^m_{j,i} \end{aligned} \tag{26}$$

$$\partial L / \partial b = -(y'_{m+1,i} - y_{m+1,i}) \times f'_m = -\delta^{m+1}_j \tag{27}$$

where m is the m-th layer of SAEs; $y_{m,i}$ is the expected values; $y'_{m,i}$ is the current values; f is the sigmoid function. The back propagation of the residual error δ is

$$\delta^m_i = \delta^{m+1}_j \times W^m_{j,i} \times f'_m \tag{28}$$

D. Algorithm Description

The Q-learning-based SAEs algorithm flow chart is shown in Fig. 3.

In the interaction process with the environment, we will not train the SAE model in the first certain time because we have no optimal actions. With time, increasingly more optimal action will be found and stored in the state-action look-up table. Then, the SAE model will be trained according to the sufficiently optimal state action information. Therefore, when the system transfers into the hidden state, we use the trained SAE network to map between the state and the action and find the optimal action. Then, we execute the new action and update the look-up table. In the follow-up time, if the system transfers to the learned state, the relay will query the state-action table to obtain the executable action.

V. ALGORITHM COMPARISONS

Now, we compare the complexity of each algorithm, as shown in table 1. In the system, the number of buffers is *K*, the buffer length is *L*, the number of channels is *M*, the number of channel states is *C*, and the number of possible transmission modes is *J*. Accordingly, the number of system states is $S = (L+1)^K \times C^M$. To each system state, the possible action number is $A = K \times M \times (J+1)$. The system scale is $D = S \times A$.

To verify the performance of the algorithm, we compare it with the strategy iteration (SI) algorithm [33], W learning (WL) algorithm [32] and random selection (RS) algorithm.

1) Strategy iteration algorithm

We can use the SI algorithm to obtain the optimal action as described in formula (37).

$$V_{n+1} = \max_{a \in A}[r(a) + \gamma p(a) V_n] = r(a_n) + \gamma p(a_n) V_n \tag{29}$$

The SI algorithm must know the system state transition probability information. However, when the system scale is notably massive, the number of linear equations is equal to the system states (i.e., $S = (L+1)^K \times C^M$). Thus, the calculation would be notably massive. It is notably hard to solve because of the problem of dimensional disaster. Therefore, the SI algorithm is not practical in wireless networks.

2) W Learning algorithm

In the W learning method, the Q learning method is used to obtain the *Q* value; then, the obtained value is used for W learning. The *W* value represents the difference between the expected return and the actual return.

$$\begin{aligned} &W_{i+1}(s_i) = \\ &\qquad (1-\alpha) W_i(s_i) + \alpha (Q_i(s_i, a_i) - (r_i + \gamma \max_{a_i \in A} Q_i(s_{i+1}, a_i))) \end{aligned} \tag{30}$$

3) Random Selection algorithm

The random selection algorithm is that each system frame randomly selects an action to execute.

In the iteration process, the SI algorithm prefers to calculate all states of the system in one iteration time, whereas the proposed algorithm in this paper prefers to calculate according to the current state. The difference of the RS algorithm is the method of action selection. The RS algorithm randomly selects an action to perform with no extra calculation, so its computational overhead is small. For the Q learning phrase, the calculation overhead of the algorithm in this paper is equivalent to that of the W learning algorithm and lower than that of the SI algorithm.

TABLE 1

ALGORITHM COMPLEXITY COMPARISON

Algorithm	Exponential Operation	Multiplication and Division	Addition and Subtraction	Comparison Operation
SI	0	$D+S$	D	$S \times A$
QL	A	$4A$	$2A$	A
WL	$2A$	$3A$	$3A$	A
RS	0	1	0	0

For the storage overhead, the SI algorithm must calculate the system state transition probability, and the storage overhead is $3(L+1)^K \times C^M$. The storage in the strategy calculation phase is $(L+1)^K \times C^M$, whereas the proposed method does not need the prior information storage overhead, and the Q learning storage overhead is $K \times M \times (J+1)$, which is much less than SI. The deep learning model requires some storage overhead. However, the model structure in this paper is relatively simple. Therefore, in general, the storage overhead of the proposed method is not larger than SI.

VI. SIMULATION EXPERIMENT AND ANALYSIS

A. Simulation Setting

In the simulation, we define the parameters of the system as follows. The number of wireless communication pairs is $K = 3$, so the number of buffers is $K = 3$. The length of each buffer is $L = 5$. The number of channels is $M = 2$. The number of channel states is $C = 4$. The possible transmission modes are no transmission, QPSK, 4-QAM, 8QAM, 16QAM, therefore $j \in \{0, 2, 4, 8, 16\}$ and $J = 4$. Coding rate V is 2. For the problem, we compared the three described algorithms with the algorithm of this paper. The simulation parameters were set as shown in table 2. We use the SAE model with two hidden layers. Further performance comparisons are shown in part B.

The SAEs with many hidden layers cannot be fully trained when the data quantity is small. More errors are also introduced

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

because there are more hidden layers. Therefore, the number of hidden layers is selected according to the simulation comparison.

TABLE 2

SIMULATION PARAMETER SETTING

Parameters	Value/Description
Threshold of SNR /*dB*	$snr = [-6.28, -1.28, 1.28]$
Doppler frequency shift /*Hz*	$f_d = 50$
Frame length /*s*	$T_f = 2\times10^{-3}$
Slot number I_1	$I_1 = 5\times10^3$
Slot number I_2	$I_2 = 1\times10^3$
Noise power WN_0/W	1×10^{-3}
Buffer pressure coefficient	$\theta = 0.5$
Arrival rate	$\lambda = [0.1, \cdots, 0.9]$
BER constraint	$BER \le 10^{-3}$
Discount coefficient	$\gamma = 0.9$
Index weight	$C_p = 1/\sqrt{2}$
MQL Learning rate	$\alpha \in (0,1]$
SAEs hidden neuron number	[8,15,15,9]
Weight	$\mu = 3\times10^{-3}$
SAEs Learning rate	$\beta = 1\times10^{-2}$
Training error accuracy	$rate = 1\times10^{-5}$

B. Performance Comparison

In slots I_1, the system learns the optimal actions that correspond to the system states. The optimal state action information is stored in the look-up table. Then, we train the SAE model with the obtained information and use the trained model to map the state and action in the next I_2 phrase. Figure 4 shows the Q value changing curve in the process of action iteration selection using the QL algorithm.

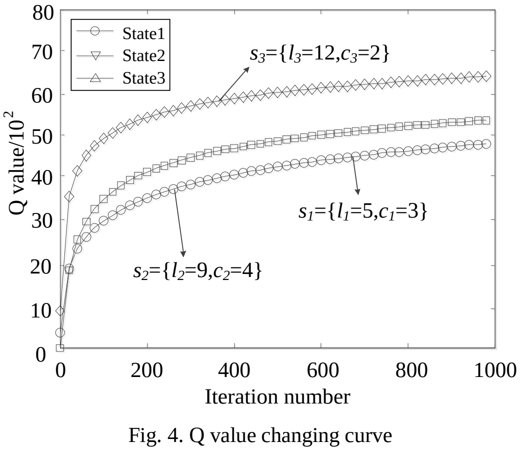

Fig. 4. Q value changing curve

Fig. 4 shows that when $\lambda = 0.1$, in states s_1={l_1=5,c_1=3}、s_2={l_2=9,c_2=4} and s_3={l_3=12,c_3=2}, the curve in the modified Q learning algorithm changes value. From Fig. 4, we observe that the three different curves converge to different values because in different system states, the existing number of packets in buffer, amount of coming packets and selection of the transmission mode are different. Therefore, the three values converge to different values, and the convergence of the algorithm is illustrated.

In the following, we compare its performance with other algorithms and use QL-SAEs to name our algorithm.

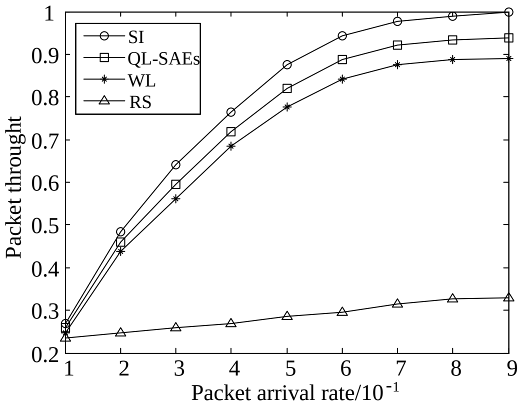

Fig. 5. Normalized throughput comparison of the algorithms

Fig. 5 shows that the system normalized throughput in one frame varies with different packet arrival rates. The QL-SAEs algorithm has a less throughput than the SI algorithm but more than the other two algorithms. This graph shows that when the packet arrival rate is small, SI, QL-SAEs and WL algorithms have almost equivalent throughputs. Because the buffer pressure is relatively small, in the learning process, the energy consumption is small (Fig. 6) in exchange for the packet throughput. When the packet arrival rate increases, the pressure of the buffer gradually increases. Then, the average amount of packet throughput in each frame gradually increases. Relatively, the arrival rate has less effect on the RS algorithm, and it can obtain a larger throughput only when there are sufficient packets in buffer and better transmission mode is selected.

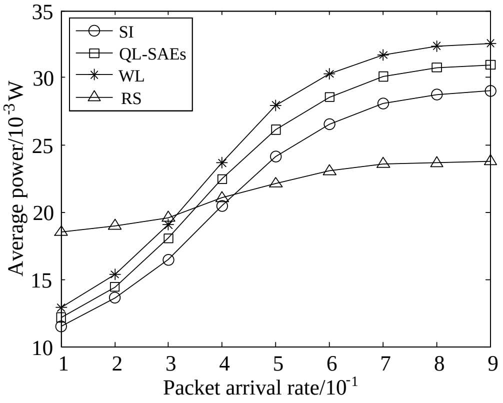

Fig. 6. Average power comparison of the algorithms

Fig. 6 shows the average power contrast of each algorithm at different packet arrival rates. When the number of packets is large, the buffer pressure is high, which can force the relay to select better transmission mode and channel to transmit more packets to reduce the buffer pressure. Ultimately, more power is consumed. The energy consumption of the three algorithms rapidly increased at first and subsequently slowly increased because with the increase of packet in the buffer, more packets can be transmitted, and more power is consumed. Therefore, the curve rapidly increases. The buffer will not have much effect on the action when the buffer reaches its limitation.

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

Finally, the power curve tends to gently increase. The power of the RS algorithm is relatively stable because the packet arrival rate basically does not affect the transmission mode selection, whereas the other three algorithms are greatly affected by λ. The other three algorithms consider the buffer pressure and transmission power, but the curses are above the curse of the RS algorithm.

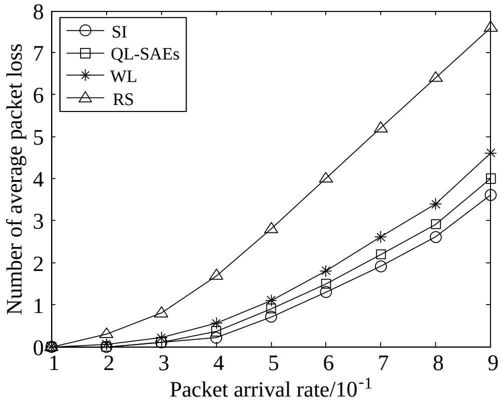

Fig. 7. Number of average packet loss of the algorithms

When the system buffer space is small, in the next frame, more packet will be possibly lost if more packets are arriving. As shown in Fig. 7, with the increase in packet arrival rate, the growth of the packet loss curves is approximately linear. Because the RS algorithm does not consider the power consumption and buffer pressure during the selection process, its packet loss is large compared to other algorithms.

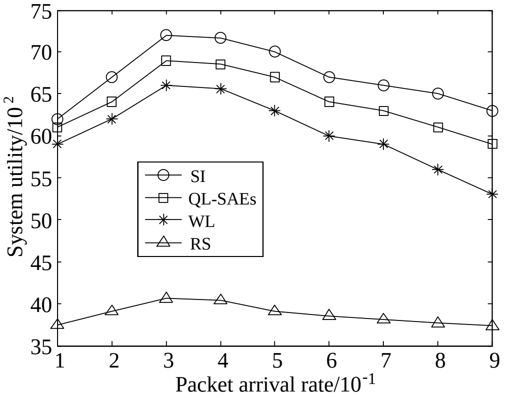

Fig. 8. Average system utility of the algorithms

TABLE 3
NOTATIONS IN THIS PAPER

Notation	Description	Notation	Description
T_f	Length of frame	$p_C(c,c^{'})$	Channel state transition probability
ρ	Ratio of Signal-to-Noise	f_d	Maximum Doppler shift
C	Channel state space	WN_0	Noise power
$p_C(c_n)$	Channel state probability	λ	Data arrival rate
S	System state space	L	Length of a signal buffer
$p_l(l,l^{'})$	Buffer state transition probability	d	Number of arriving packets
a_i	The selected action	t	Number of transmitted packets
p_{BER}	Bit error rate restriction	$P(c_i, j)$	Power consumption
j	The selected modulation	A	Action set
B	Buffer space	$p_s(s_i, s_{i+1}\|a_i)$	System transition probability
V	Coding rate	K	Number of buffers
$B(s_i,a_i)$	System benefit	$f_{k,i}$	Pressure of buffer *k*
θ	Buffer pressure coefficient	$C(s_i,a_i)$	System cost
O_i	System utility	$V^{\pi}(s_i)$	System long term reward
$Q(s_i,a_i)$	Evaluation of the state and action	γ	Discount coefficient
$Index(s_i,a)$	Action index	$T(n)$	Selection number of the action
$V_a(i)$	Bias factor	$\sigma_i^2(n)$	Action utility value variance
α	Learning rate	*Output*	Output vector of SAEs

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Notation	Description	Notation	Description
$Input$	Input vector of SAEs	n_i	Neuron number in SAEs input layer
M	Number of available channels	n_o	Neuron number in SAEs output layer
Con	A constant limited in [1,10]	n_k	Neuron number in SAEs hidden layer
$L(X)$	Cost function	x^i	Input data
J_{W^l}	Weight decay term	$f(x^i)$	Output data
μ	Weight decay term coefficient	W_{ji}^l	Weight
β	Learning rate	δ	Residual error

The average utility of the system is shown in Fig. 8. The graph shows that the utilities of SI, QL-SAEs and WL algorithm are higher than RS. Although QL-SAEs have a lower system utility value than SI, it is better than the WL algorithm. To constrain the buffer space, the utility of the system is not notably large when the arrival rate of the packet is too large or too small. When λ is small, the system can select the appropriate action to enhance the utility value. However, when the amount of packet is large, although the system is attempting to transmit packets, it cannot completely transmit all packets. In addition, the transmission power consumption is notably large if there are many packets.

VII. CONCLUSION

In this paper, a relay for transmission packets to the sink of the other nodes is considered in the CIoT. To solve the problem of transmission scheduling in the CIoT, a new Q-learning-based transmission scheduling mechanism using deep learning has been proposed to achieve the appropriate strategy with multiple channels for the cognitive node. In this paper, the reinforcement learning, which is based on the joint action selection criteria of the index value and *Q* value to balance the action equilibrium problem in exploration and exploitation, is used to solve the MDP problem. Ultimately, it realizes the long-term maximum utility of the system. In addition, the SEA deep learning model is introduced to map between the state and the action. Although the proposed scheme has a lower performance than SI, the algorithm complexity is strongly reduced. The proposed algorithm can work without priori information, which is suitable for practical scenarios.

The research in this paper is based on the case of one relay. However, if there are more than one relay scenario, the method to make many relays cooperatively or competitively work can be further studied.

REFERENCES

[1] Luca Mainetti, et al. Evolution of wireless sensor networks towards the Internet of Things: A survey [C]. IEEE International Conference on Software, Telecommunications and Computer Networks, September 2011: 15-17.

[2] Khalil N, Abid M R, Benhaddou D, et al. Wireless sensors networks for Internet of Things[C]// IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. IEEE, 2014:1-6.

[3] S. Jeschke, et al. (Eds.), "Industrial Internet of Things: Cyber-manufacturing Systems," ISBN 978-3-319-42559-7, Springer, Switzerland, 2016.

[4] Lin Y, Yang J, Lv Z, et al. A self-assessment stereo capture model applicable to the Internet of Things[J]. Sensors, 2015, 15(8): 20925-20944.

[5] Yang J, He S, Lin Y, et al. Multimedia cloud transmission and storage system based on Internet of Things[J]. Multimedia Tools and Applications, 2015: 1-16.

[6] Maw H, Xiao H, Christianson B, et al. BTG-AC: Break-The-Glass Access Control Model for Medical Data in Wireless Sensor Networks.[J]. IEEE Journal of Biomedical & Health Informatics, 2015, 20(3):763-774.

[7] Yuan B, Fu C, Chen D. Building a Large Scale Wireless Sensor Network for the Industrial Environment[C]// IEEE, International Conference on Embedded and Real-Time Computing Systems and Applications. IEEE Computer Society, 2016:96-96.

[8] Ma Z, Pan X. Agricultural environment information collection system based on wireless sensor network[C]// Global High Tech Congress on Electronics. IEEE, 2012:24-28.

[9] M. Chen, J. Yang, Y. Hao, S. Mao, K. Hwang, "A 5G Cognitive System for Healthcare", Big Data and Cognitive Computing, Vol. 1, No. 1, DOI:10.3390/bdcc1010002, 2017.

[10] Min Chen, Y. Hao , K. Hwang, L. Wang, L. Wang, "Disease Prediction by Machine Learning over Big Healthcare Data", IEEE Access, DOI: 10.1109/ACCESS.2017.2694446, 2017.

[11] Tervonen J, Mikhaylov K, Pieska S, et al. Cognitive Internet-of-Things solutions enabled by wireless sensor and actuator networks[C]// IEEE CogInfoCom. IEEE, 2014:97-102.

[12] J. Lloret, et al, (Eds.), Cognitive Networks: Applications and Deployments, ISBN: 1482236990 | 978-1482236996, CRC Press - Taylor & Francis Group, USA, 2014.

[13] Wu Q, Ding G, Xu Y, et al. Cognitive Internet of Things: A New Paradigm Beyond Connection[J]. 2014, 1(2):129-143.

[14] Khan A A, Rehmani M H, Rachedi A. When Cognitive Radio meets the Internet of Things?[C]// International Wireless Communications and Mobile Computing Conference. 2016:469-474.

[15] Shah M A, Zhang S, Maple C. Cognitive radio networks for Internet of Things: Applications, challenges and future[C]// International Conference on Automation and Computing. 2013:1-6.

[16] Haustein, Thomas, et al. "Cognitive Wireless Communications–A paradigm shift in dealing with radio resources as a prerequisite for the wireless network of the future–An overview on the topic of cognitive wireless technologies." Frequenz 70.7-8 (2016): 281-288.

[17] Otermat D T, Kostanic I, Otero C E. Analysis of the FM Radio Spectrum for Secondary Licensing of Low-Power Short-Range Cognitive Internet of Things Devices[J]. IEEE Access, 2016, PP(99):1-1.

[18] Somov A, Dupont C, Giaffreda R. Supporting smart-city mobility with cognitive Internet of Things[C]// Future Network and Mobile Summit. 2013:1-10.

[19] Nitti M, Murroni M, Fadda M, et al. Exploiting Social Internet of Things Features in Cognitive Radio[J]. IEEE Access, 2016, PP(99):1-1.

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

[20] Zhu J, Song Y, Jiang D, et al. Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the Internet of Things[J]. IEEE Access, 2016, 4:4609-4617.

[21] Maghsudi S, Stanczak S. Hybrid Centralized-Distributed Resource Allocation for Device-to-Device Communication Underlaying Cellular Networks[J]. IEEE Transactions on Vehicular Technology, 2016:1.

[22] Yang J, Chen B, Zhou J, et al. A low-power and portable biomedical device for respiratory monitoring with a stable power source[J]. Sensors, 2015, 15(8): 19618-19632.

[23] D. Jiang, Z. Xu, Jindi Liu, et al. An optimization-based robust routing algorithm to energy-efficient networks for cloud computing[J]. Telecommunication Systems, March 14, 2015, online available.

[24] Sanchez S M, Souza R D, Fernandez E M G, et al, Rate and energy efficient power control in a cognitive radio Ad hoc network[J]. IEEE Signal Processing Letters, 2013, 20(5): 451-454.

[25] Aniba G, Aissa S. Cross-Layer Designed Adaptive Modulation Algorithm with Packet Combining and Truncated ARQ over MIMO Nakagami Fading Channels[J]. IEEE Transactions on Wireless Communications, 2011, 10(4):1026-1031.

[26] Gao Q, Zhu G, Lin S, et al. Robust QoS-Aware Cross-layer Design of Adaptive Modulation Transmission on OFDM Systems in High-Speed Railway[J]. IEEE Access, 2016, PP(99):1-1.

[27] Ramachandran B, Shanmugavel S. Mobility Adaptive Cross Layer Design for Reliable Route Discovery in Ad-hoc Networks[C]// International Conference on Wireless Communication and Sensor Networks. 2007:69 - 73.

[28] Chen J, Lv T, Zheng H. Joint Cross-Layer Design for Wireless QoS Content Delivery[J]. EURASIP Journal on Advances in Signal Processing, 2005, 2005(2):1-16.

[29] Praveena T, Nagaraja G S, Shashank C S, et al. A novel Scheduling Algorithm emphasizing fairness for Cross Layer Design in wireless networks[C]// 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). 2016:210-215.

[30] Dong Y, Dong H. Simulation study on cross-layer design for energy conservation in underwater acoustic networks[C]// Oceans. IEEE, 2013:1-5.

[31] Zhu J, Xu B Y, Li S Q. Optimal and Suboptimal Access and Transmission Polices for Dynamic Spectrum Access over Fading Channels in Cognitive Radio Networks[J], Chinese Journal of Electronics,2008, 17(4): 726-73.

[32] Zhu J, Peng Z, Li F. A transmission and scheduling scheme based on W-learning algorithm in wireless networks[C]// Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on. IEEE, 2013: 85-90.

[33] LIN Xiao-hui, TAN Yu, ZHANG Jun-ling. MDP-based energy efficient policy for wireless transmission[J]. Systems Engineering and Electronics. 2014, 36(7):1433-1438.

[34] Xu Zheng, Zhipeng Cai, Jianzhong Li and Hong Gao. A Study on Application-aware Scheduling in Wireless Networks. IEEE Transactions on Mobile Computing, 2017 , 16 (7) :1787-1801

[35] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao.Scheduling Flows with Multiple Service Frequency Constraints.IEEE Internet of Things, 2017 , 4 (2) :496-504.

[36] Zhuojun Duan, Wei Li and Zhipeng Cai. Distributed Auctions for Task Assignment and Scheduling in Mobile Crowdsensing Systems.The 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017)

[37] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. An Application-aware Scheduling Policy for Real-time Traffic. The 35th IEEE International Conference on Distributed Computing Systems (ICDCS 2015), 2015 :421-430.

[38] WANG H S，MOAYERI N．Finite- state Markov channel-a useful model for radio communication channels[J]. IEEE Transactions on Vehicular Technology，1995，44(1) : 163-171

[39] Qian Gao, Gang Zhu, Siyu Lin, et al. Robust QoS-Aware Cross-Layer Design of Adaptive Modulation Transmission on OFDM Systems in High-Speed Railway[J]. IEEE ACCESS. 2016, PP(99):1-1.

[40] Lau V K N. Performance of variable rate bit interleaved coding for high bandwidth efficiency[J] //Proc. Of the Vehicular Technology Congerence. 2000, 3:2054--2058.

[41] Chung S T, Goldsmith A J. Degrees of freedom in adaptive modulation: a unified view[C]// IEEE Transactions on Communications. 2001:1561 - 1571.

[42] HUMPHRYS M. W-learning: competition among selfish Q-learners [R]. Cambridge: University of Cambridge，1995．

[43] Wei Q, Liu D, Shi G. A Novel Dual Iterative, Q-Learning Method for Optimal Battery Management in Smart Residential Environments[J]. IEEE Transactions on Industrial Electronics, 2015, 62(4):2509-2518.

[44] Ni J, Liu M, Ren L, et al. A Multiagent Q-Learning-Based Optimal Allocation Approach for Urban Water Resource Management System[J]. IEEE Transactions on Automation Science & Engineering, 2014, 11(1):204-214.

[45] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search.[J]. Nature, 2016, 529(7587):484-489.

[46] Spencer M, Eickholt J, Cheng J. A Deep Learning Network Approach to abinitio Protein Secondary Structure Prediction [J]. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2015, 12(1):103-112.

[47] Huang W, Song G, Hong H, et al. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning [J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):2191-2201.

[48] Wu M, Chen L. Image recognition based on deep learning[C]// Chinese Automation Congress (CAC), 2015. IEEE, 2015.

[49] Liu W, Liu N, Sun H, et al. Dispatching Algorithm Design for Elevator Group Control System with Q-Learning based on a Recurrent Neural Network[C]// Chinese Control and Decision Conference. 2013:3397-3402.

[50] Li C, Zhang J, Li Y. Application of Artificial Neural Network Based on Q-learning for Mobile Robot Path Planning[C]// 2006 IEEE International Conference on Information Acquisition. Shandong University, 2006:978-982.

[51] Kobayashi T, Shibuya T, Tanaka F, et al. Q-learning in Continuous State-Action Space by Using a Selective Desensitization Neural Network[J]. Ieice Technical Report Neurocomputing, 2011, 111:119-123.

[52] Comsa I S, Zhang S, Aydin M. A novel dynamic Q-learning-based scheduler technique for LTE-advanced technologies using neural networks[C]// IEEE, Conference on Local Computer Networks. IEEE Computer Society, 2012:332-335.

[53] Teng T H, Tan A H. Fast Reinforcement Learning under Uncertainties with Self-Organizing Neural Networks[C]// IEEE / Wic / ACM International Conference on Web Intelligence and Intelligent Agent Technology. IEEE, 2015:51-58.

[54] Browne C B, Powley E, Whitehouse D, et al. A Survey of Monte Carlo Tree Search Methods[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2012, 4:1(1):1 - 43.

[55] WATKINS C J C H, DAYAN P. Technical Note: Q-Learning[J]. Machine Learning, 1992, 8(3):279-292.

Jiang Zhu received the Ph.D. degree in Communication and Information from university of electronic science and technology, China, in 2009. He is an associate professor of School of Telecommunications &Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His current researches include cognitive radio, wireless sensor networks and mobile communication security technology.

Yonghui Song received his B.E. degree in North China Institute of Aerospace Engineering, China, in 2014. He is currently working toward his M.E. degree in Chongqing University of Posts and Telecommunications (CQUPT). His research interests include wireless sensor networks and cognitive radio.

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2759728, IEEE Internet of Things Journal

Dingde Jiang (S'08–M'09) is currently a professor in the School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu, China. He was a professor in the School of Computer Science and Engineering and the College of Information Science and Engineering at Northeastern University, Shenyang, China, before April in 2017. He received the Ph.D. in Communication and Information Systems from University of Electronic Science and Technology of China, Chengdu, China, in 2009. From 2013 to 2014, he was a Visiting Scholar with the Department of Computer Science and Engineering at University of Minnesota, Minneapolis, MN, USA. His research focuses on network measurement, modeling and optimization, performance analysis, network management, network security in communication networks, particularly in software defined networks, information-centric networking, energy-efficient networks, and cognitive networks.

His research is supported by the National Science Foundation of China, the Program for New Century Excellent Talents in University of Ministry of Education of China, and so on. He has been serving as an Editor for 1 international journals. He served as a technical program committee (TPC) member for several international conferences. He has received the Best Paper Awards at several International Conferences.

Houbing Song (M'12–SM'14) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in August 2012.

In August 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). He served on the faculty of West Virginia University from August 2012 to August 2017. In 2007 he was an Engineering Research Associate with the Texas A&M Transportation Institute. He serves as an Associate Technical Editor for IEEE Communications Magazine. He is the editor of four books, including Smart Cities: Foundations, Principles and Applications, Hoboken, NJ: Wiley, 2017, Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications, Chichester, UK: Wiley-IEEE Press, 2017, Cyber-Physical Systems: Foundations, Principles and Applications, Boston, MA: Academic Press, 2016, and Industrial Internet of Things: Cybermanufacturing Systems, Cham, Switzerland: Springer, 2016. He is the author of more than 100 articles. His research interests include cyber-physical systems, cybersecurity and privacy, internet of things, edge computing, big data analytics, unmanned aircraft systems, connected vehicle, smart and connected health, and wireless communications and networking.

Dr. Song is a senior member of ACM. Dr. Song was the very first recipient of the Golden Bear Scholar Award, the highest campus-wide recognition for research excellence at West Virginia University Institute of Technology (WVU Tech), in 2016.

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.